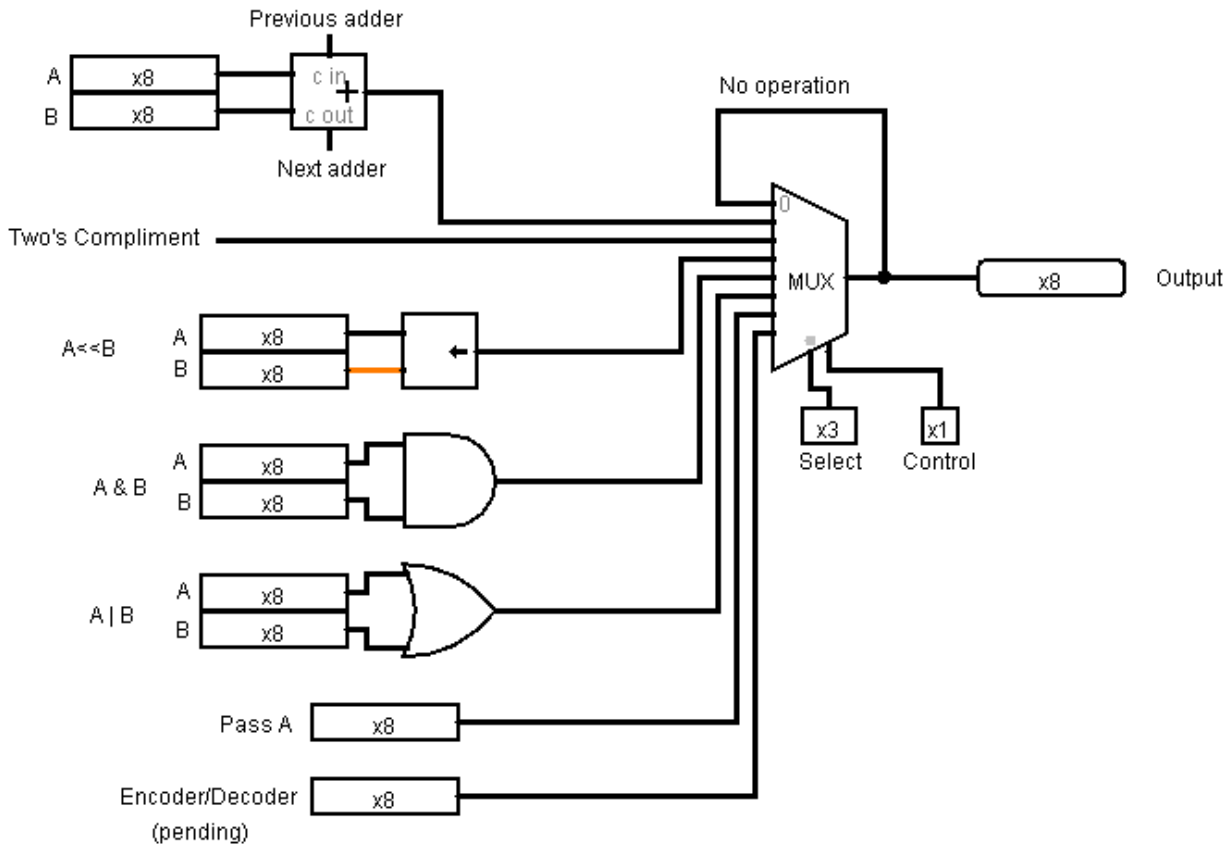
Digital Integrated Circuits - 3663
Professor T. Blalock
TA: Alicia Klinefelter

Team Gouda
Charles Moran – cwm6aw
Silvio Mancone – sfm4ex
Kelvin Green – kg9y
Kevin Lee – kjl3ac

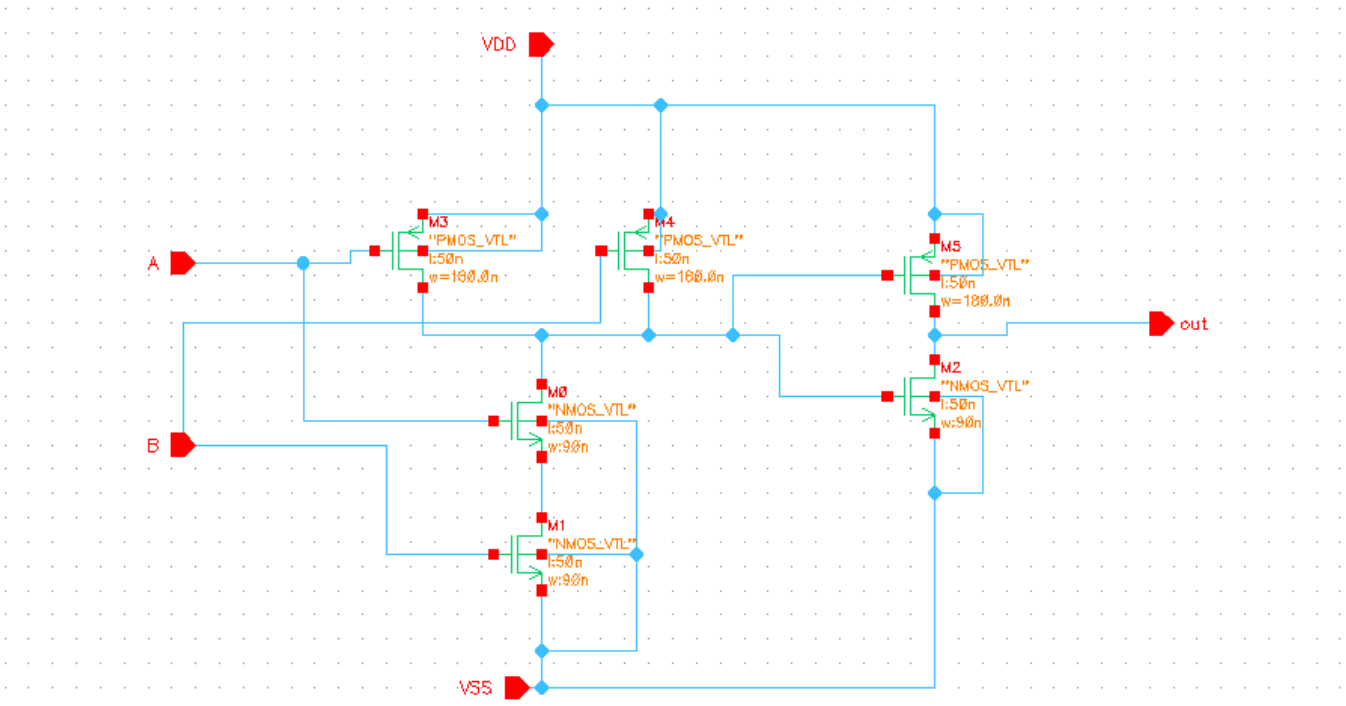
Design Project: Phase I
Due: 04/02/13, 12:30

1. Block Diagram

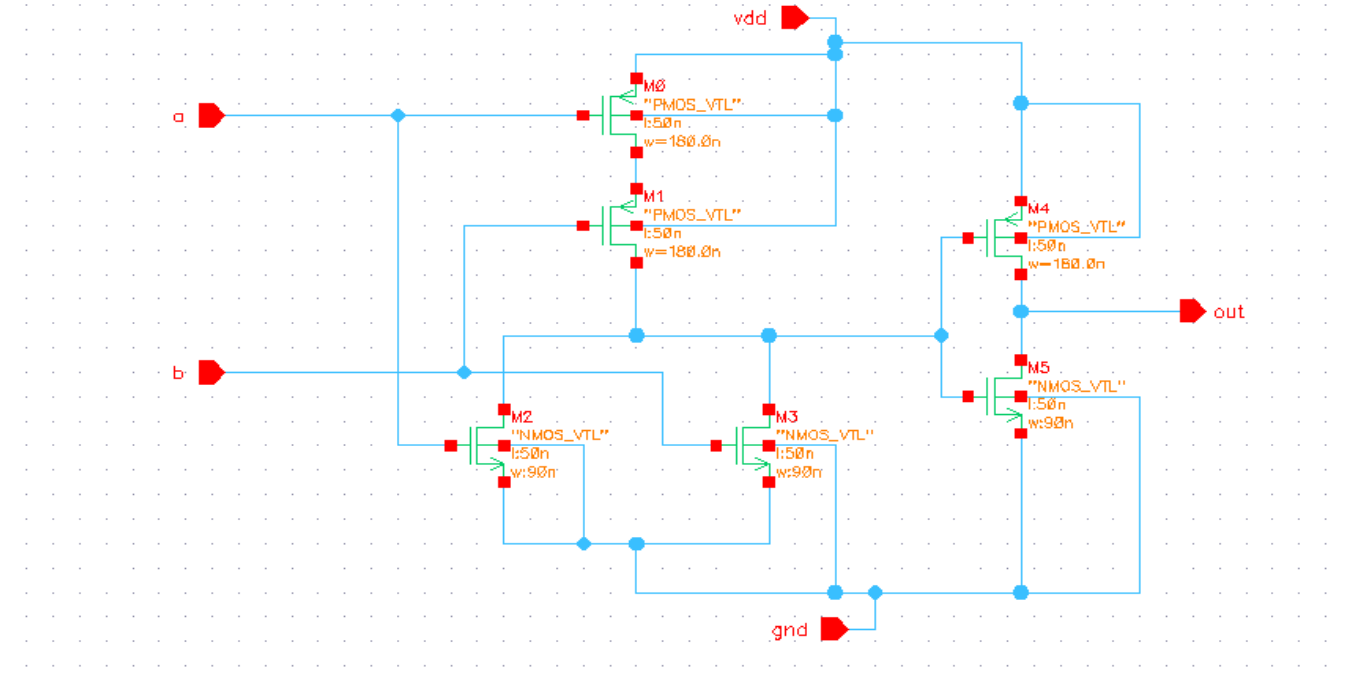


2. Component Schematics

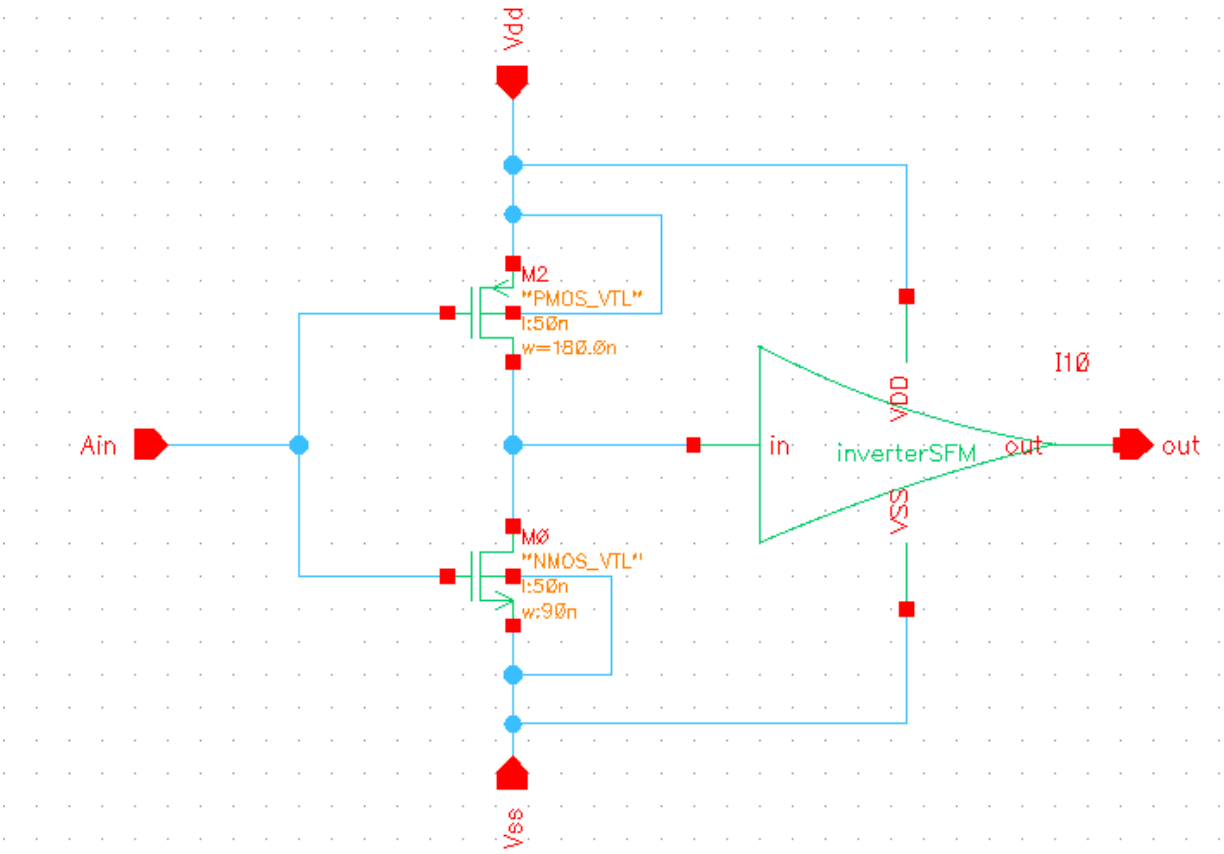
I) 2-Input AND



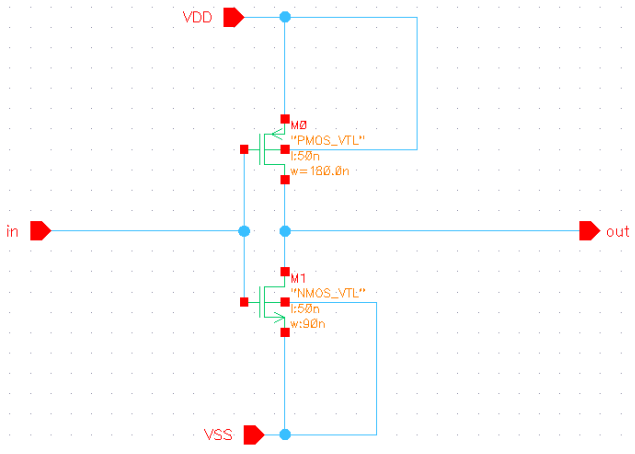
II) 2-Input OR



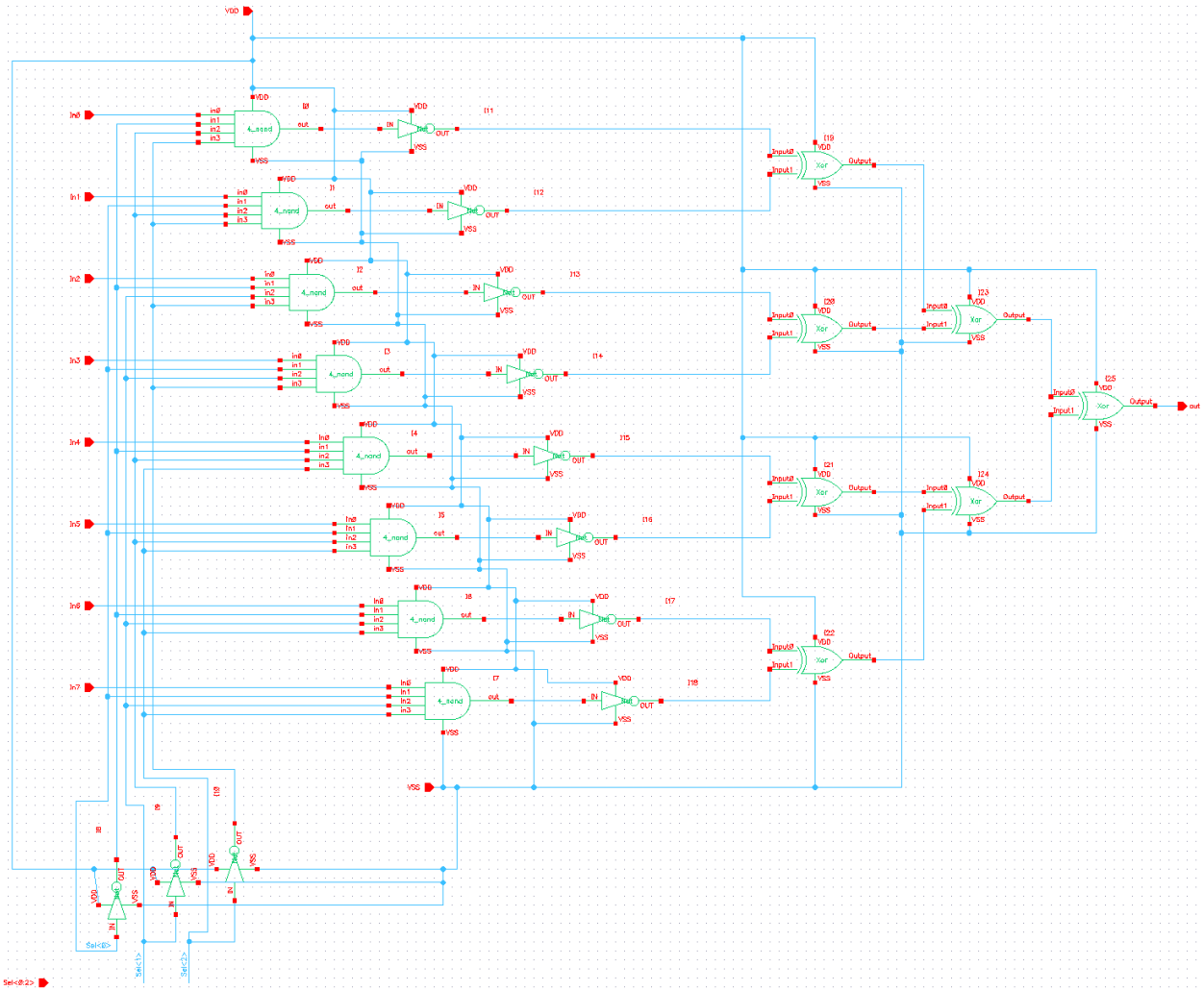
III) PASS A



Inverter:



IV) 8:1 MUX



3. Test Strategy & Plan

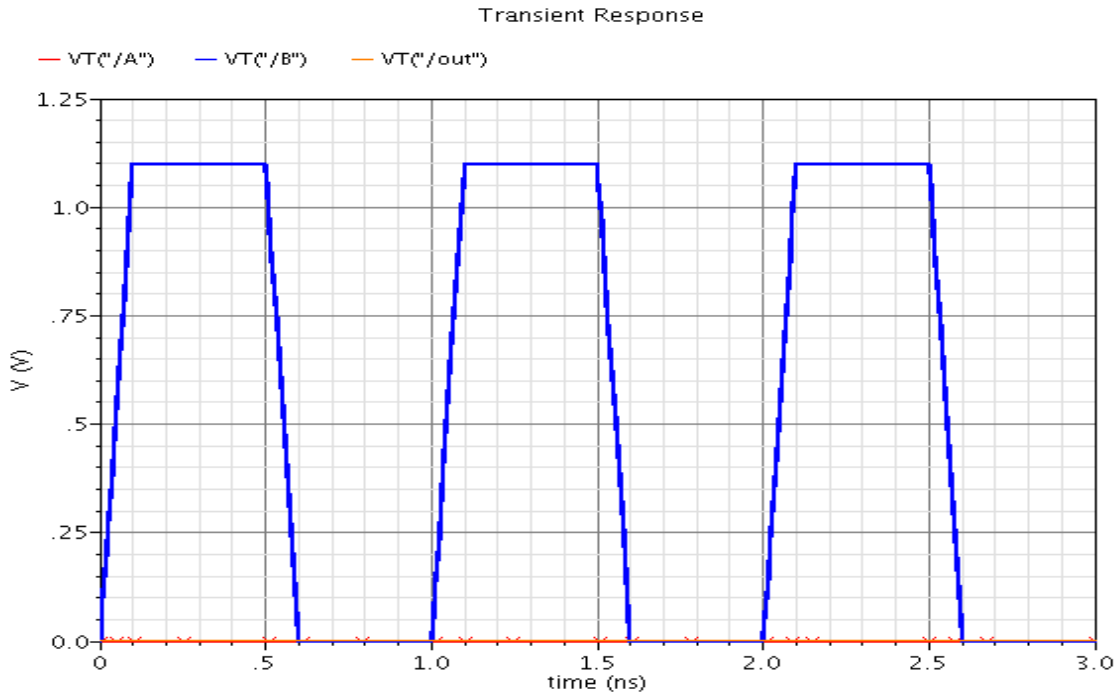
Each component was first assembled and tested on the transistor level, as instructed. While viewing the schematics on this low level, the standard approach of saving and checking was utilized, but, additionally, the Check > Find marker command was also used in an attempt to exterminate all warnings, as well as to seek a better explanation for the errors we encountered. After the components passed this level of testing, they were placed onto a test bench that each member made individually and were plotted graphically. From these plots, members verified the functionality of the components by observation.

Future testing will, hopefully, incorporate a more rigorous approach on two fronts. First, as the current way of testing the circuit's functionality rests somewhat on human error, later tests will compare the output of the schematics to those of premade chips. This will help to greatly reduce the possible error, as the circuits will not be juxtaposed to one's personal understanding of how a chip's should function. Secondly, OCEAN will later be incorporated into various aspects of design and testing. It is obvious how much more precision, accuracy, and speed could be acquired with the adoption of OCEAN. It is unfortunate that there was a bit too steep of a learning curve to adopt this technique before this phase of design was sought to be checked.

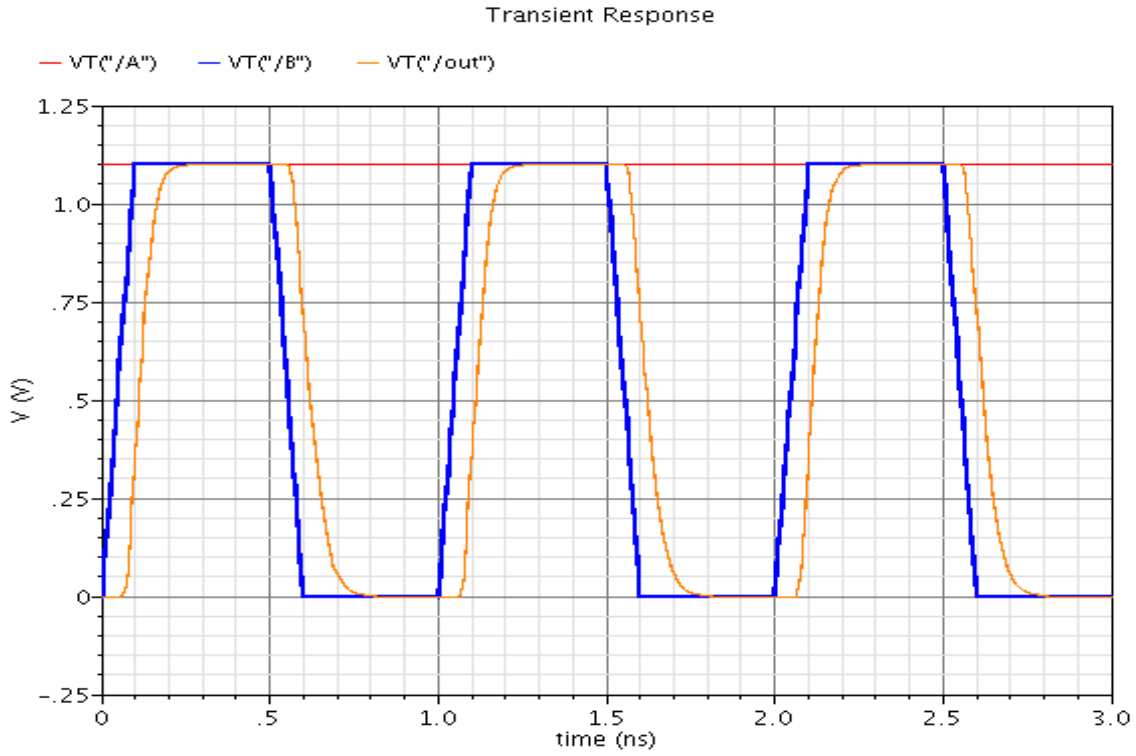
4. Simulation Results

I) 2-Inpu t AND

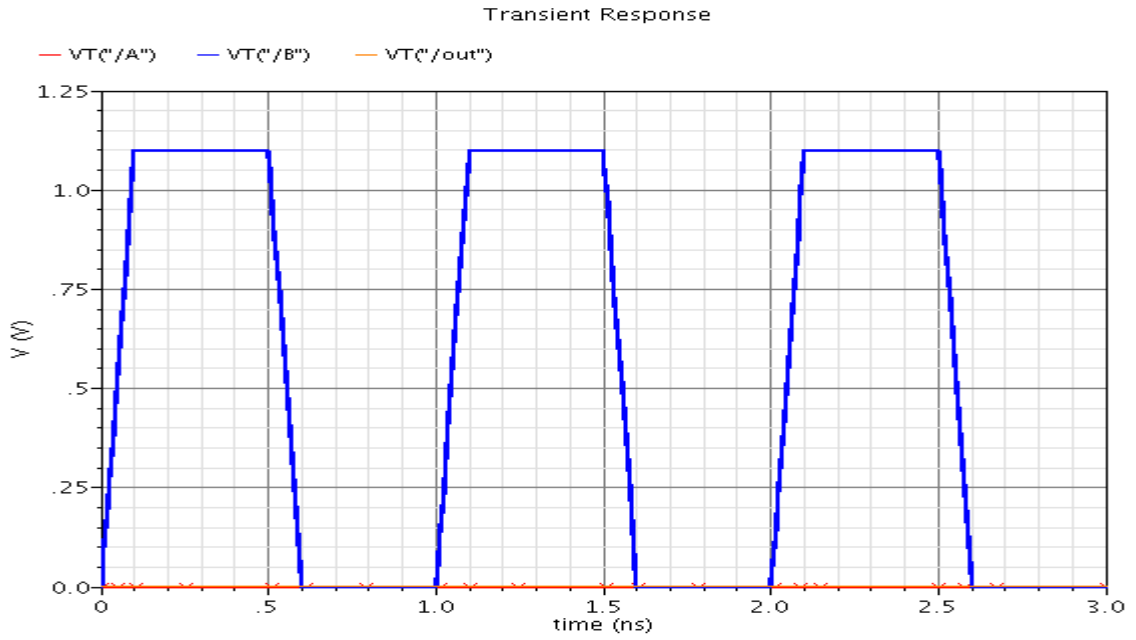
A = 0V, B = 0->1.1V Transient Signal



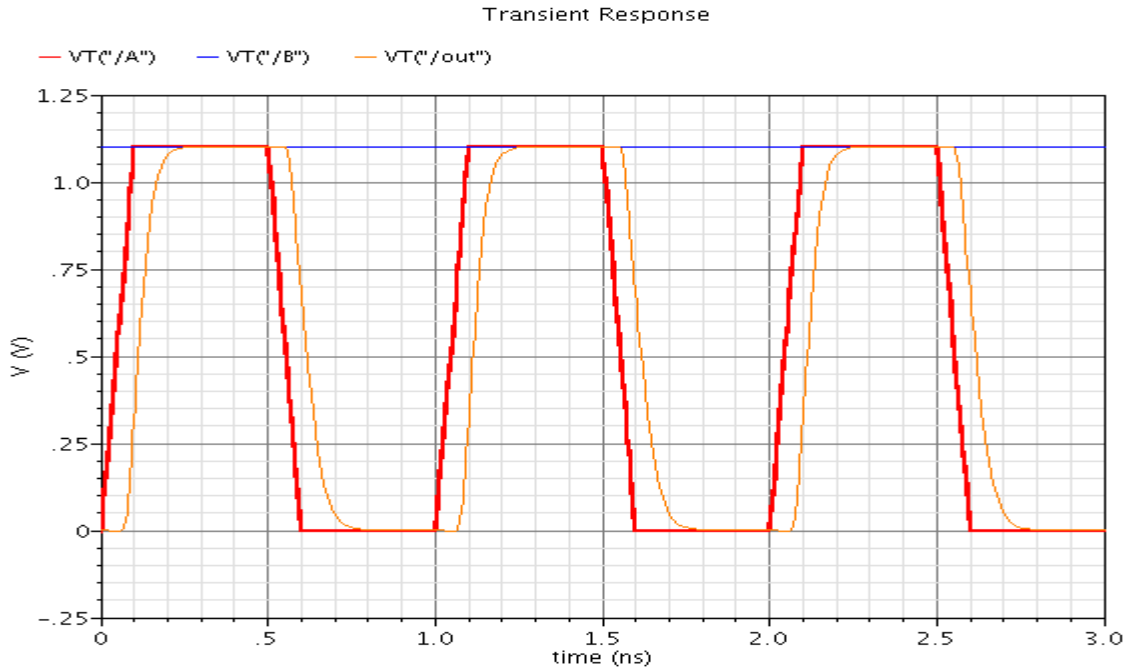
A = 1.1V, B = 0->1.1V Transient Signal



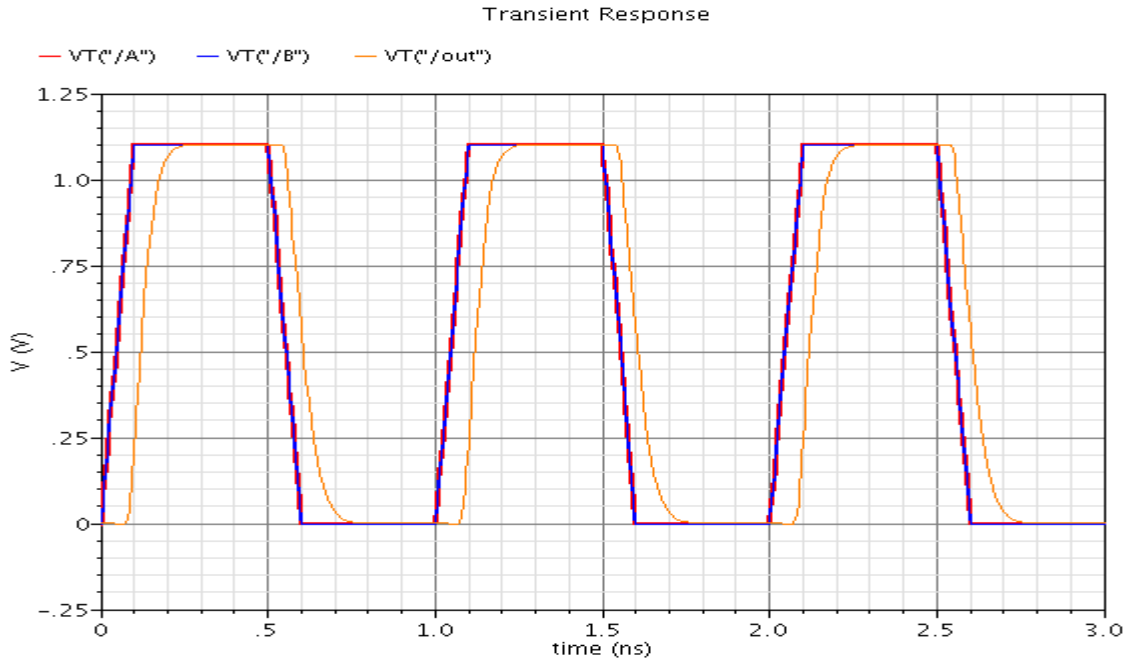
A = 0->1.1V, B = 0V Transient Signal



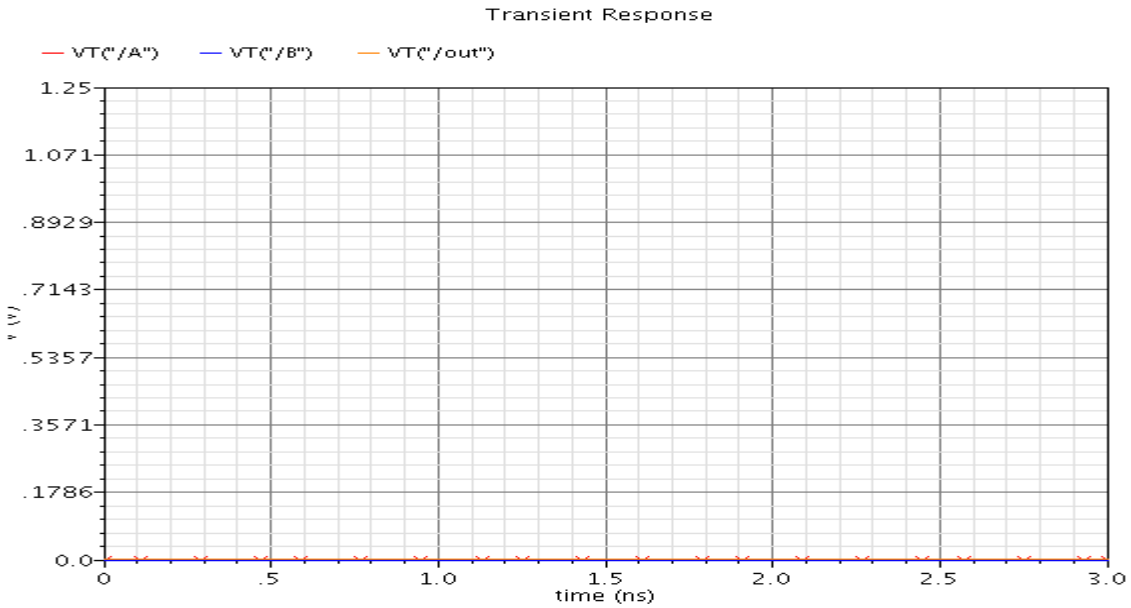
A = 0->1.1V, B = 1.1V, Transient Signal



A = 0->1.1V, B = 0->1.1V Transient Signal

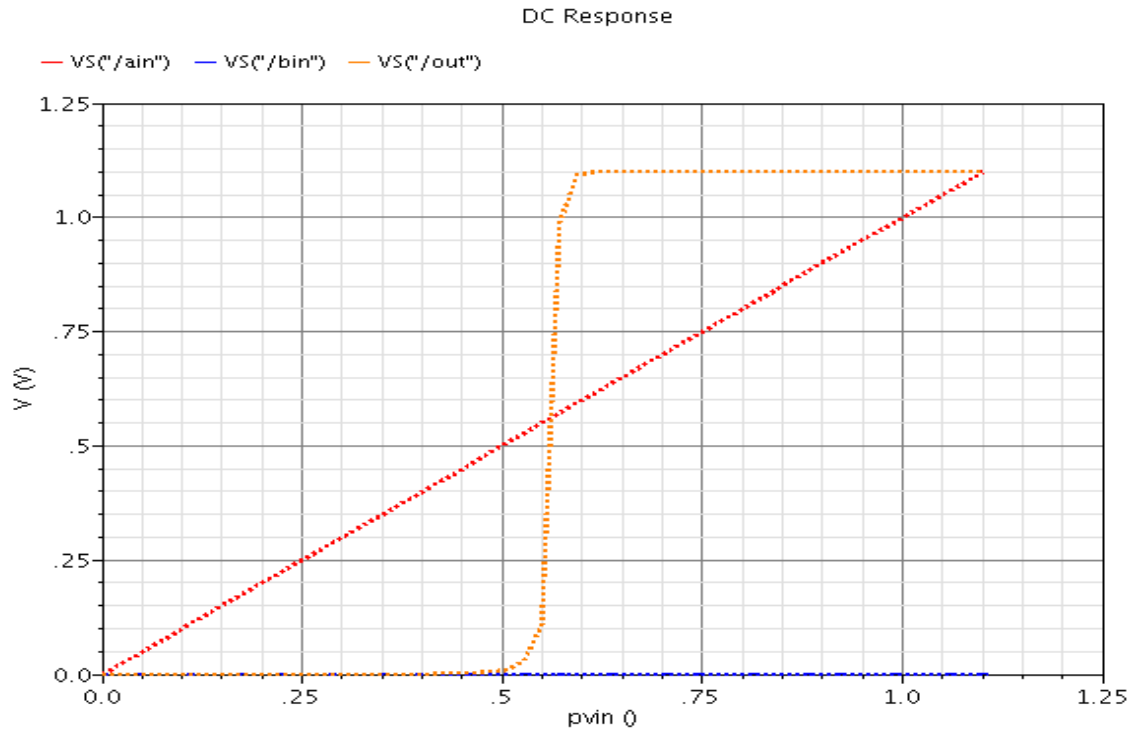


A = 0, B = 0 Transient Signal

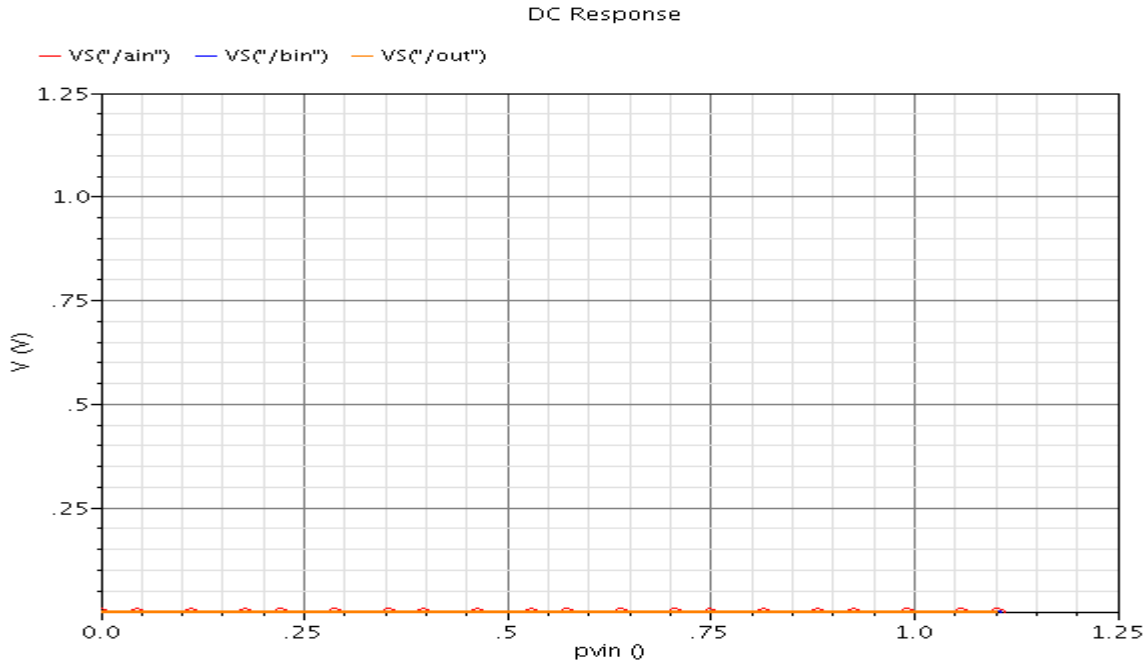


II) 2-Input OR

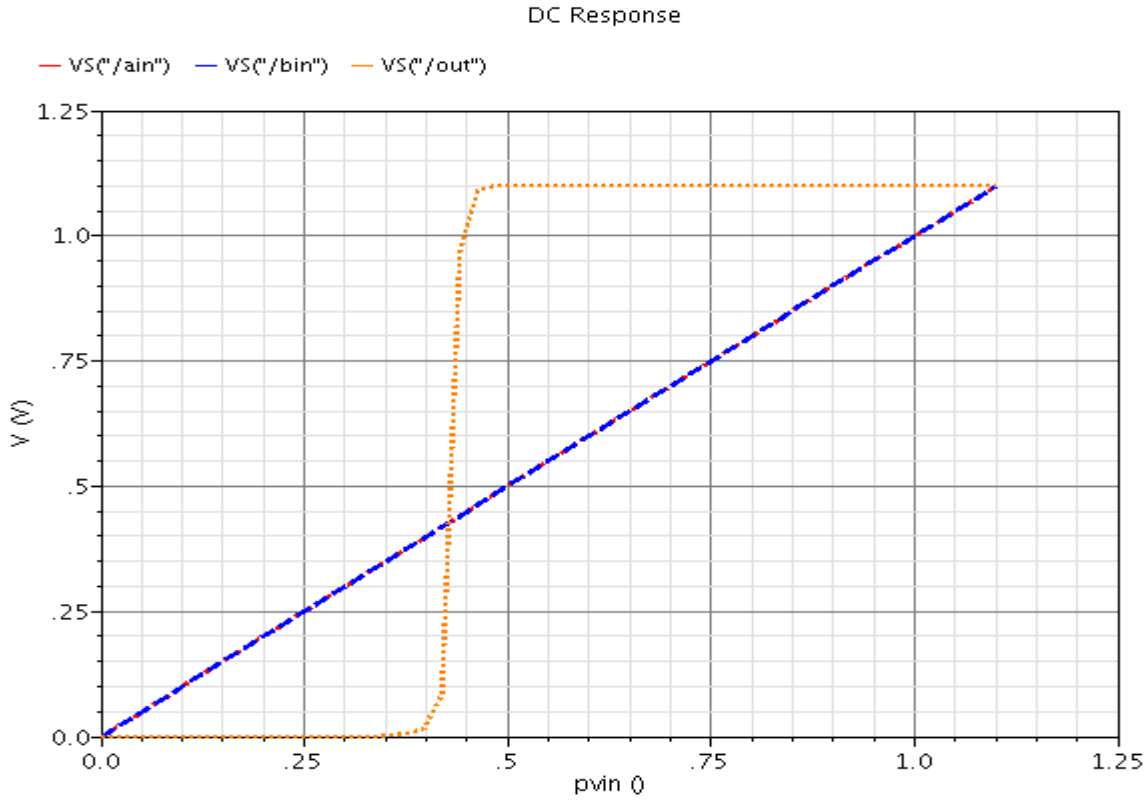
A = 0->1.1V, B = 0->1.1V



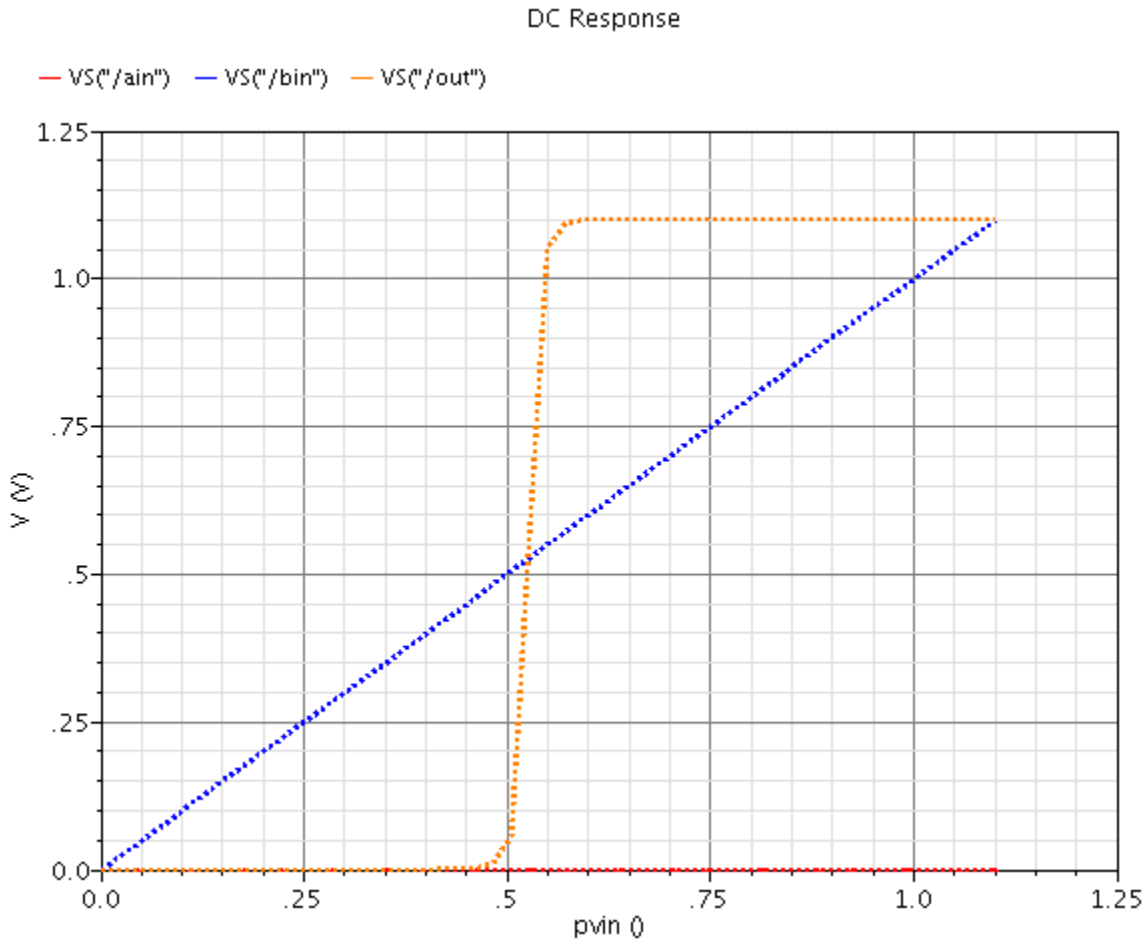
A = 0 B = 0-> A=0 B=0



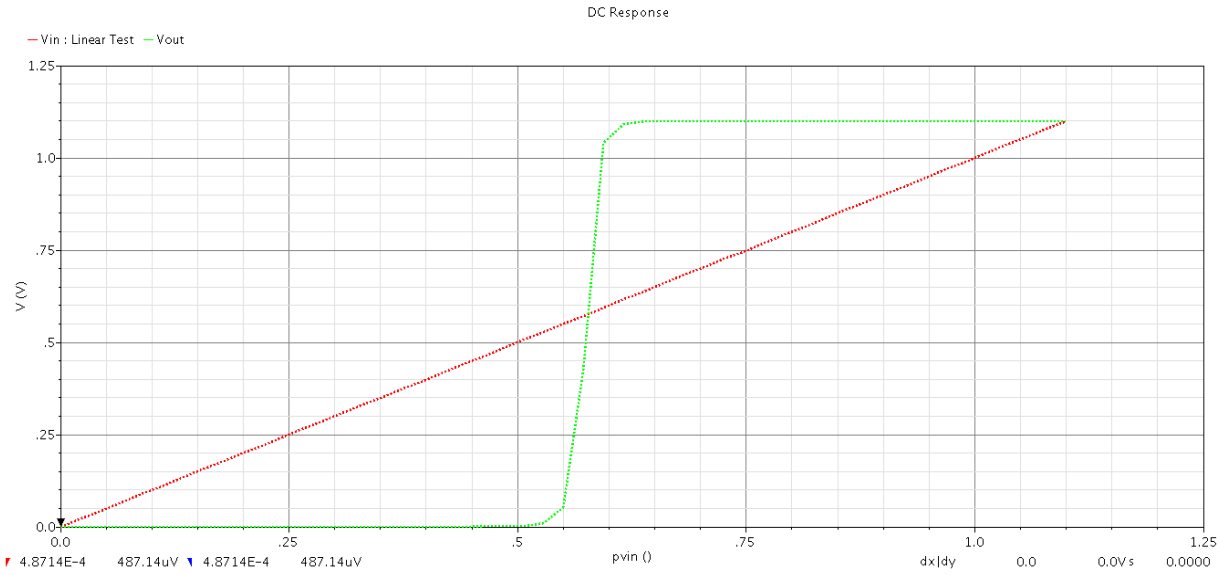
A = 0 B = 0-> A=1.1V, B=1.1V



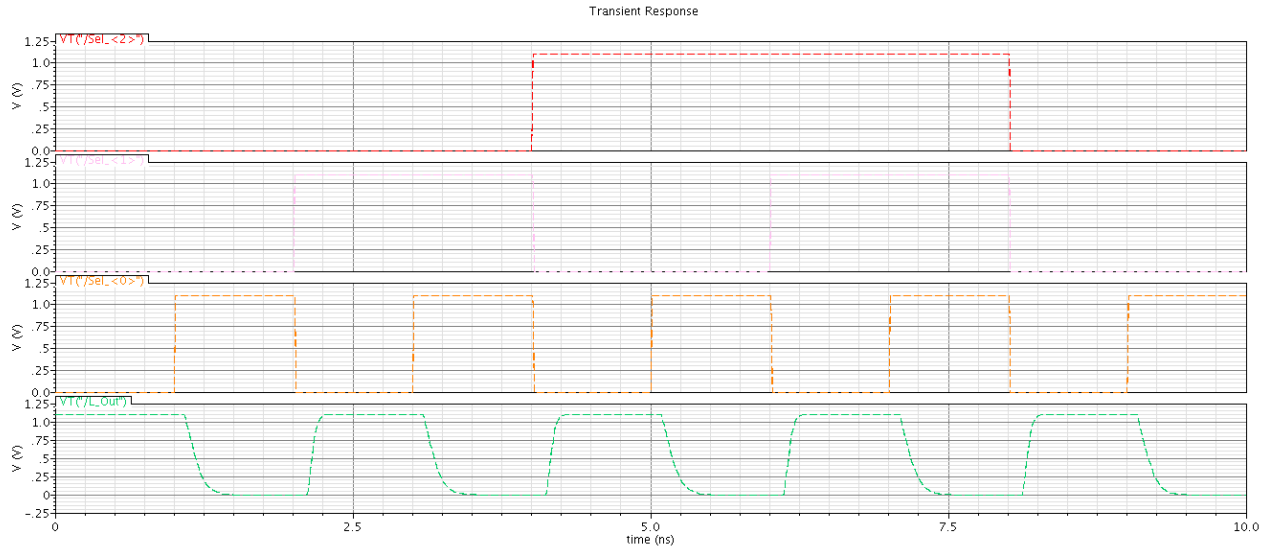
A = 0 B = 0-> A=0V, B=1.1V



III) PASS A



IV) 8:1 MUX

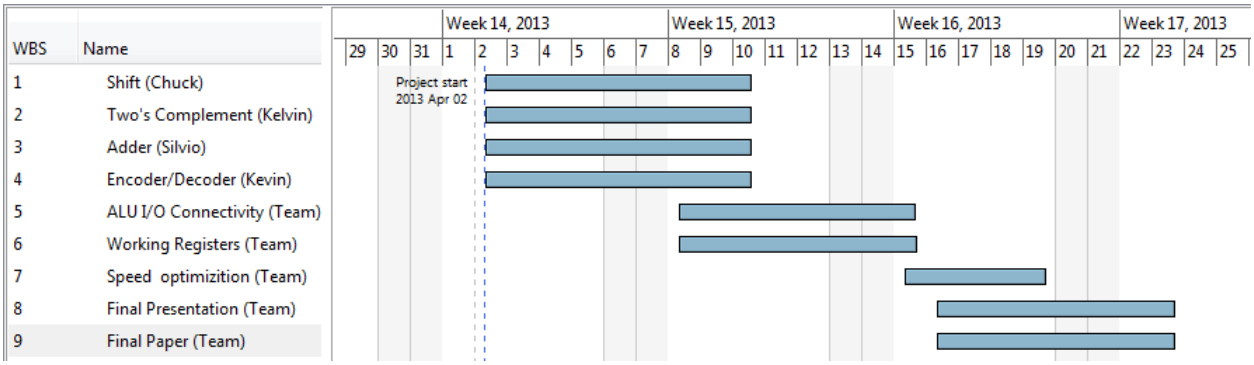


5. Progress & Future Tasks

The learning curve faced by Team Gouda was a bit steeper than expected; the first meeting went successfully, but was only comprised of assigning tasks, getting a general time layout, and of realizing the overall goal, as well as what was required to achieve this goal. The second meeting went rather slowly, as no member is particularly familiar with neither Linux nor Cadence; much time was spent by the members simply familiarizing themselves with these environments.

The initial tasks of creating the 2-Input AND, 2-Input OR, Pass A, and 8:1 MUX was accomplished within roughly 7 hours of design work per component; some of this time was spent being “lost” in Cadence or researching certain aspects of the design. The group spend a considerable amount of time brainstorming a way to design the layout so that the ALU would not have to calculate all of the possible outputs before the MUX selects which one is actually necessary.

A Work Breakdown Schedule can be gleaned from the following Gantt chart:



Lastly, the arbitrary function chosen for integration is a coder/decoder. Through transistor manipulation, it should be feasible to change a given input to a coded output, and then, after a certain number of iterations, reproduce that input as an output.